

FIGURE 1

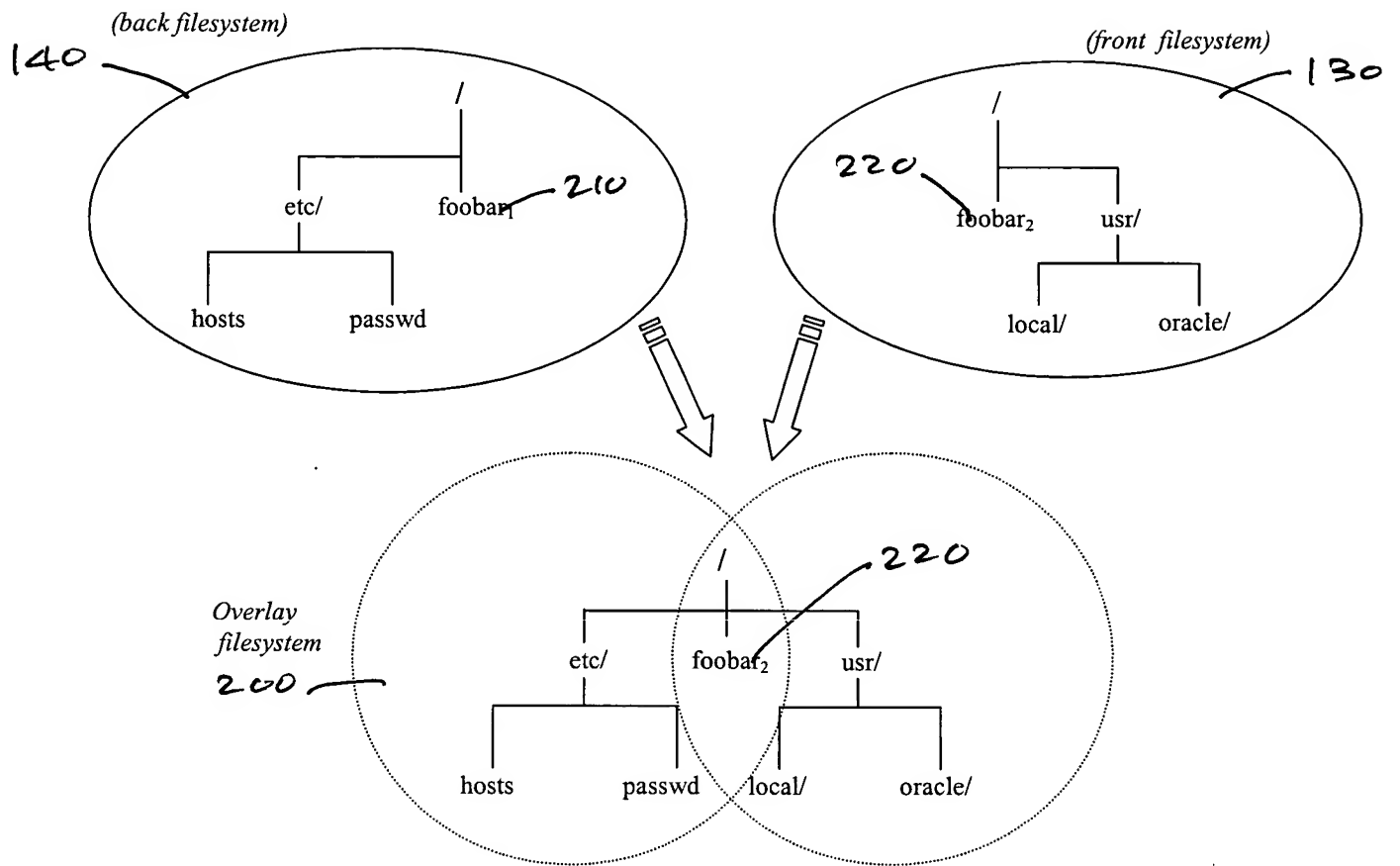


FIGURE 2

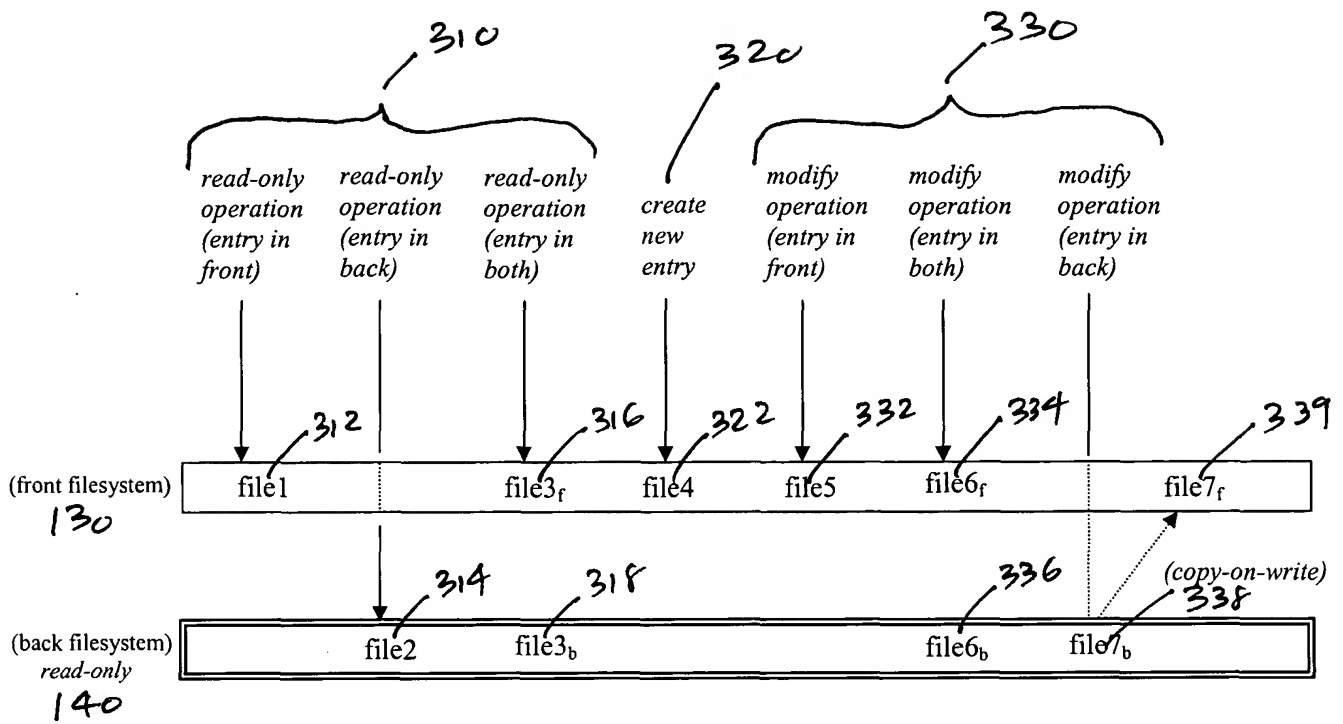


FIGURE 3

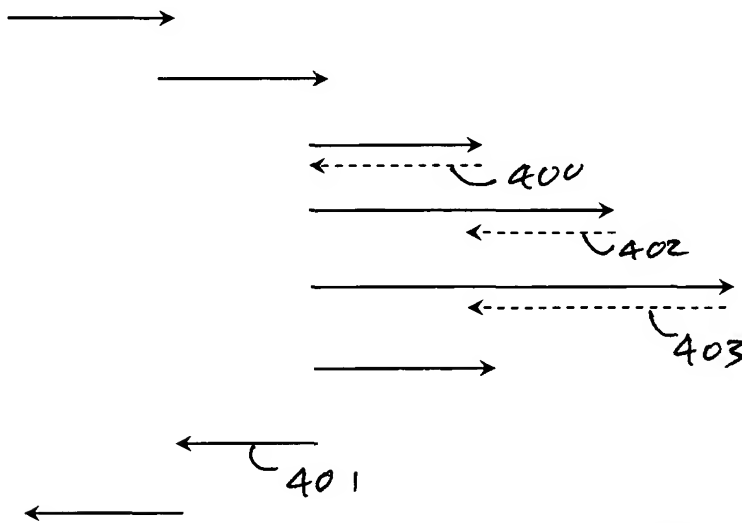
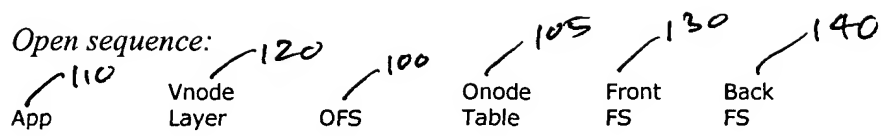


FIGURE 4A

Application issues `open()` system call to kernel.

The file is in an Ofs partition so the request is passed to Ofs.

Ofs allocates an *onode* and a *shadow vnode*.

Ofs sends the `open()` request to the front fs. The returned vnode is stored in the onode.

Ofs sends the `open()` request to the back fs. The returned vnode is stored in the onode.

The onode & shadow vnode are updated to reflect the state of the underlying vnode reference.

The onode and the shadow vnode are linked and the shadow vnode is returned to the vnode layer.

The kernel returns a file descriptor linked to the vnode back to the application.

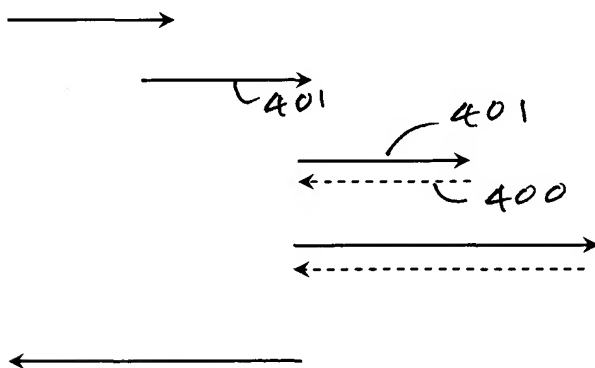
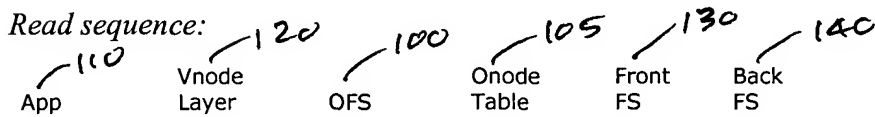


FIGURE 4B

Application issues a `read()`.

Vnode layer passes request and shadow vnode to Ofs.

Ofs uses the shadow vnode to find onode.

Ofs uses the returned onode to get the real underlying vnode and passes the `read()` request to the correct layer (front or back).

Data is returned to the application.

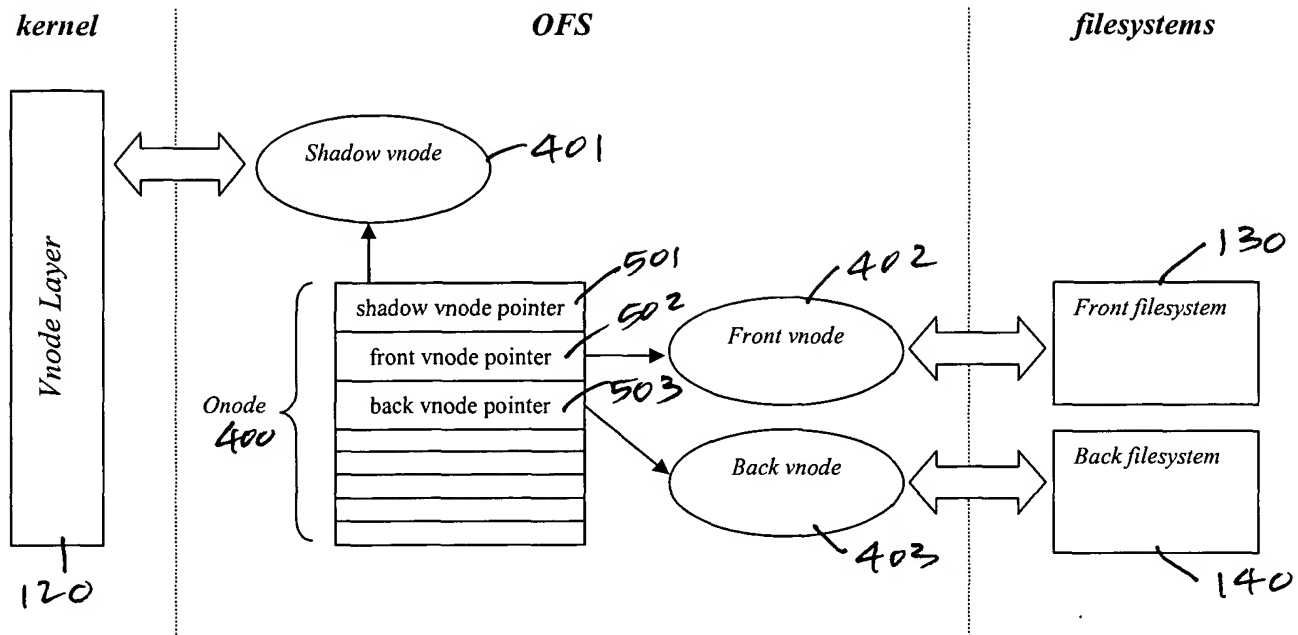


FIGURE 5

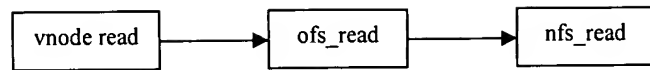


FIGURE 6

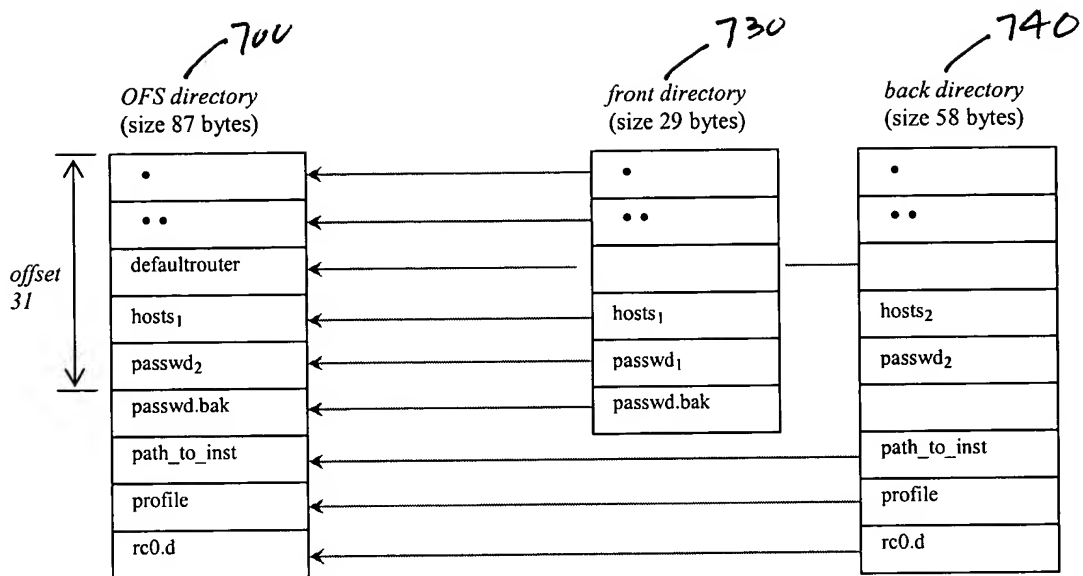


FIGURE 7

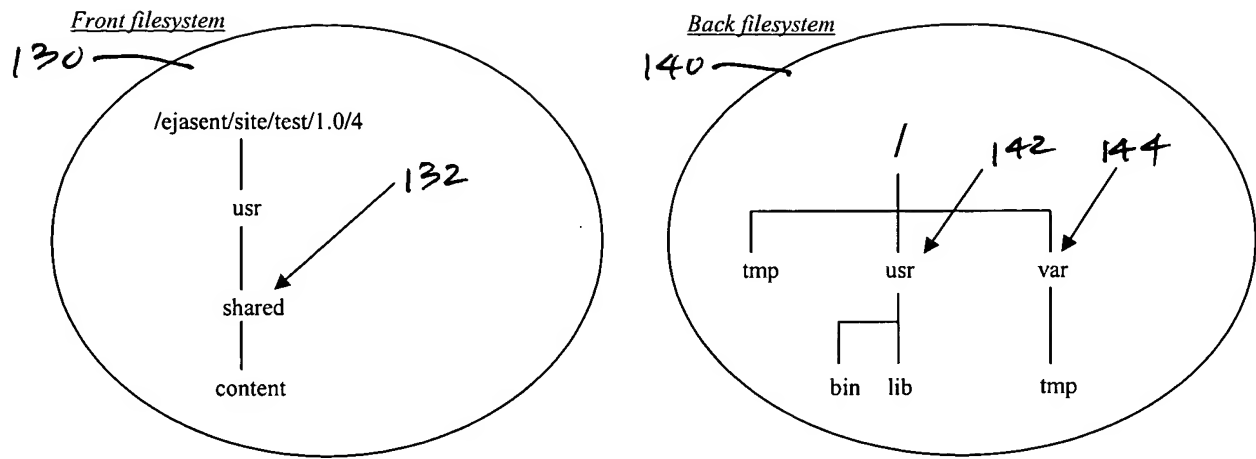


FIGURE 8A



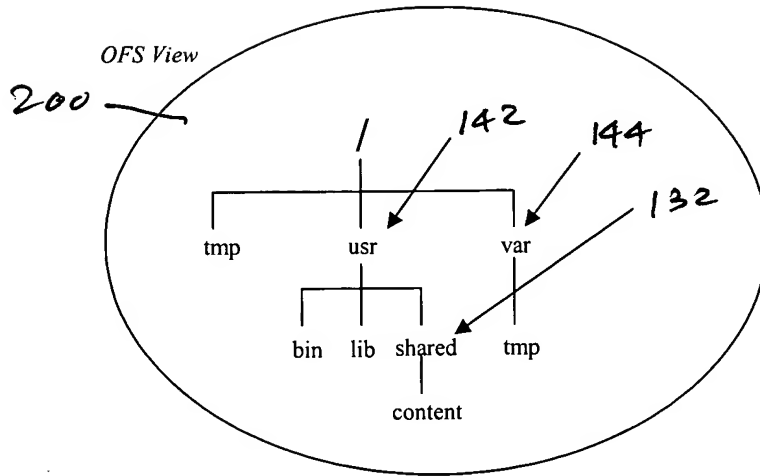


FIGURE 8B

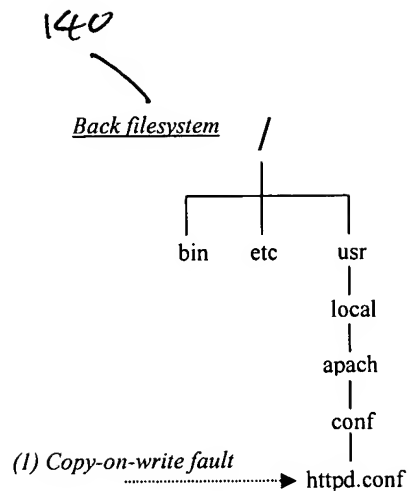


FIGURE 9A

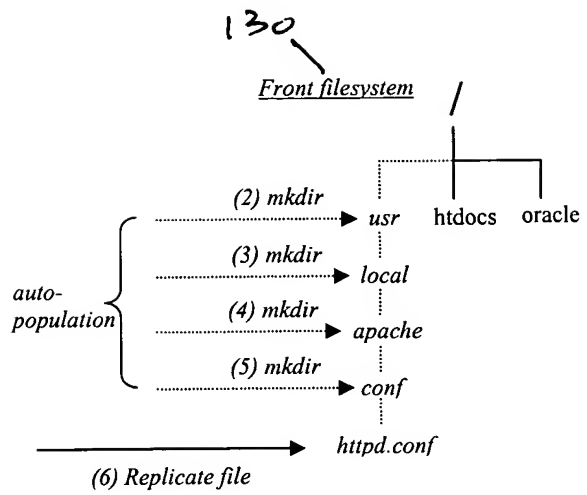


FIGURE 9B

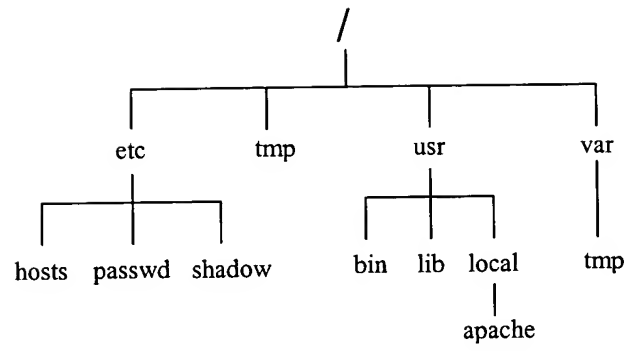


FIGURE 10

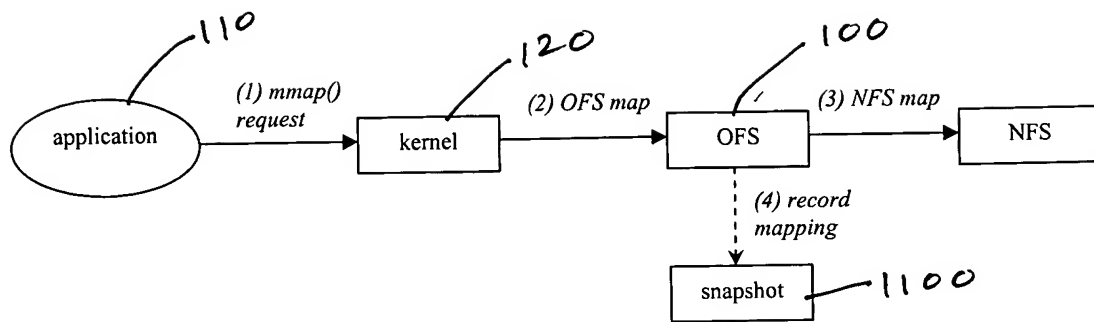


FIGURE 11

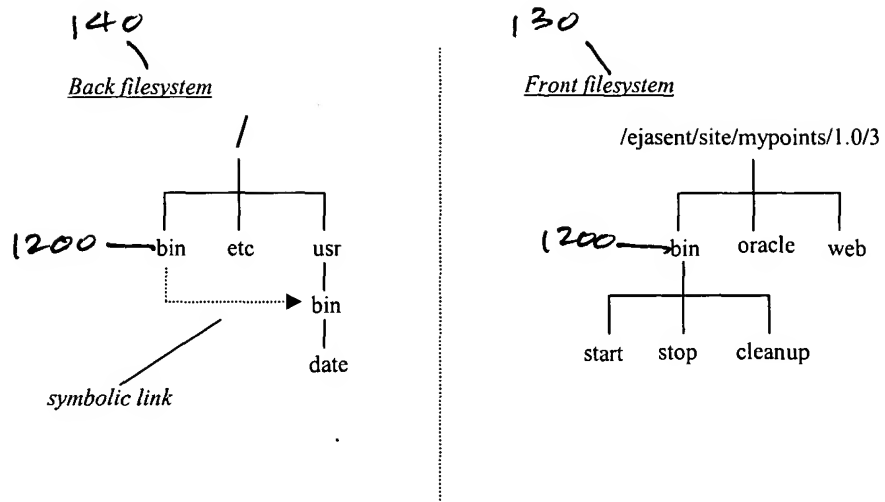


FIGURE 12

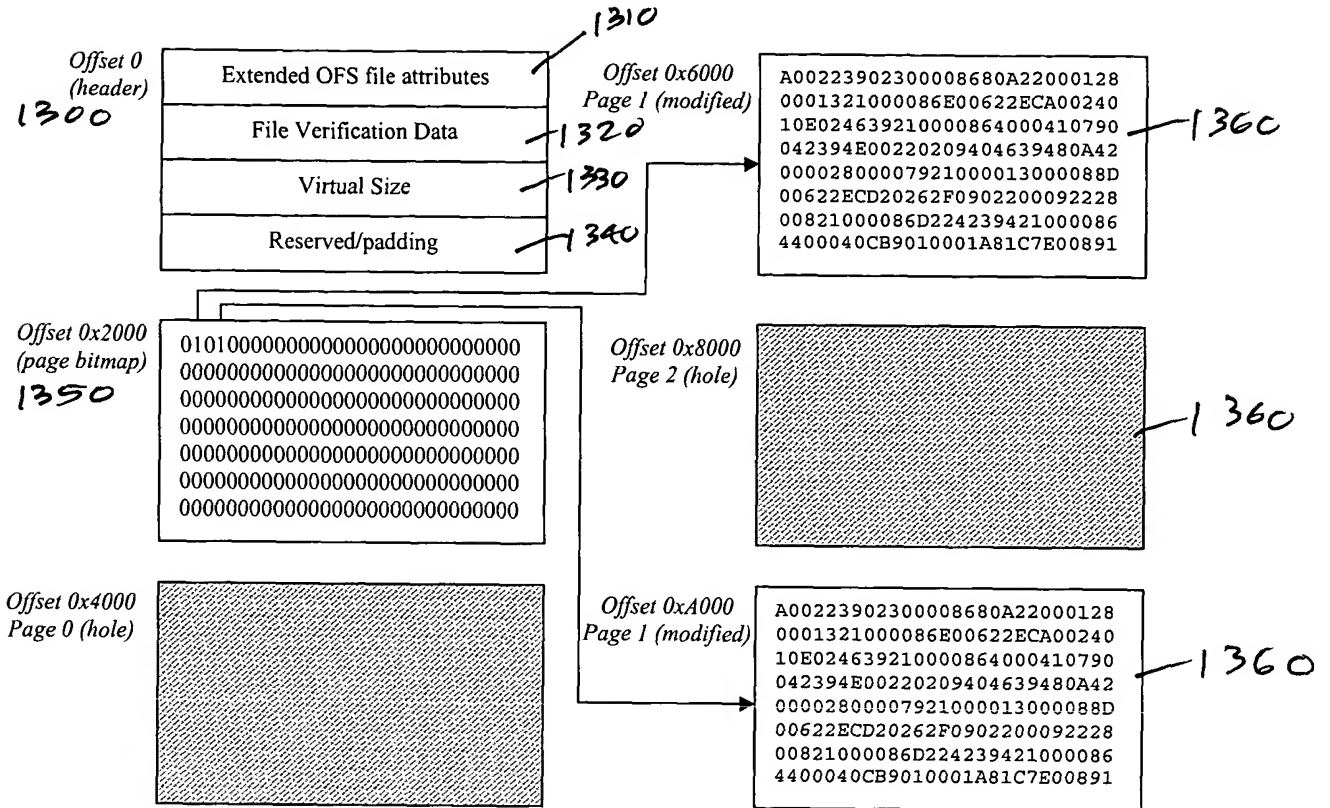


FIGURE 13A

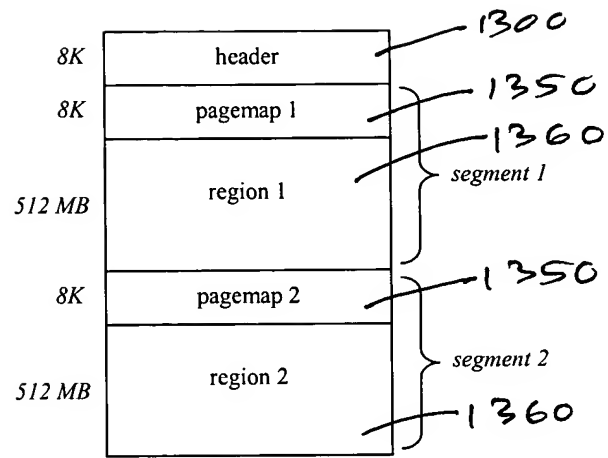


FIGURE 13B

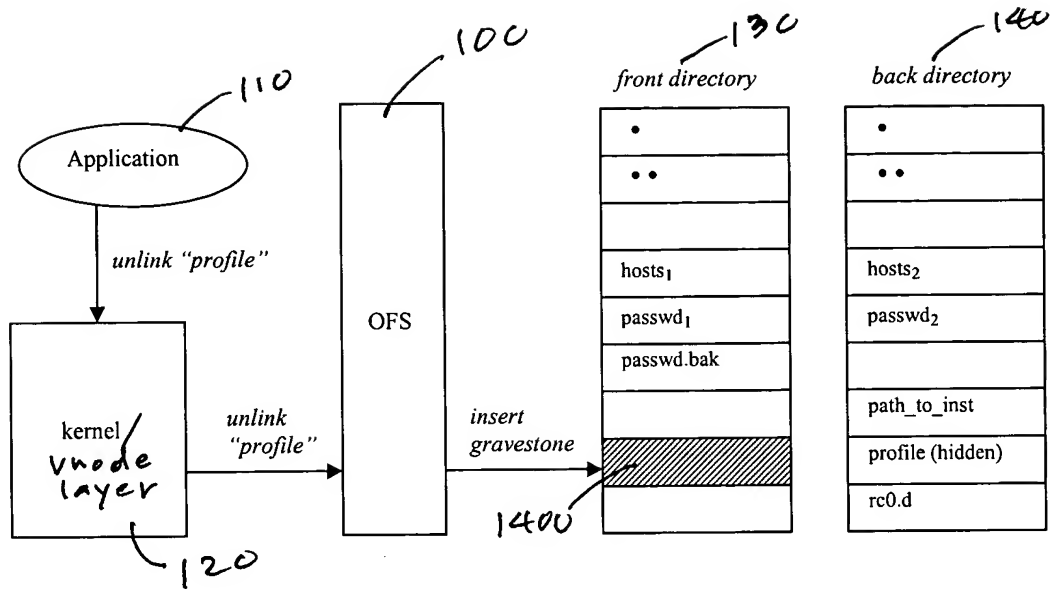


FIGURE 14